OPTi Inc.
888 Tasman Drive
Milpitas, CA 95035
(408) 486-8000
Fax: (408) 486-8001

Application Note (OPTi Confidential)

Product Name:	Viper Xpress+ Chipset
Title:	BIOS Programming Guide
Date:	January 16, 1997

Scope

This document outlines the recommended procedure for programming the internal registers of the Viper Xpress+ Chipset. It is intended to be a reference for a BIOS developer using the Viper Xpress+ Chipset in a PC design. This document classifies the internal registers of the Viper Xpress+ Chipset based on the subsystems that they control.

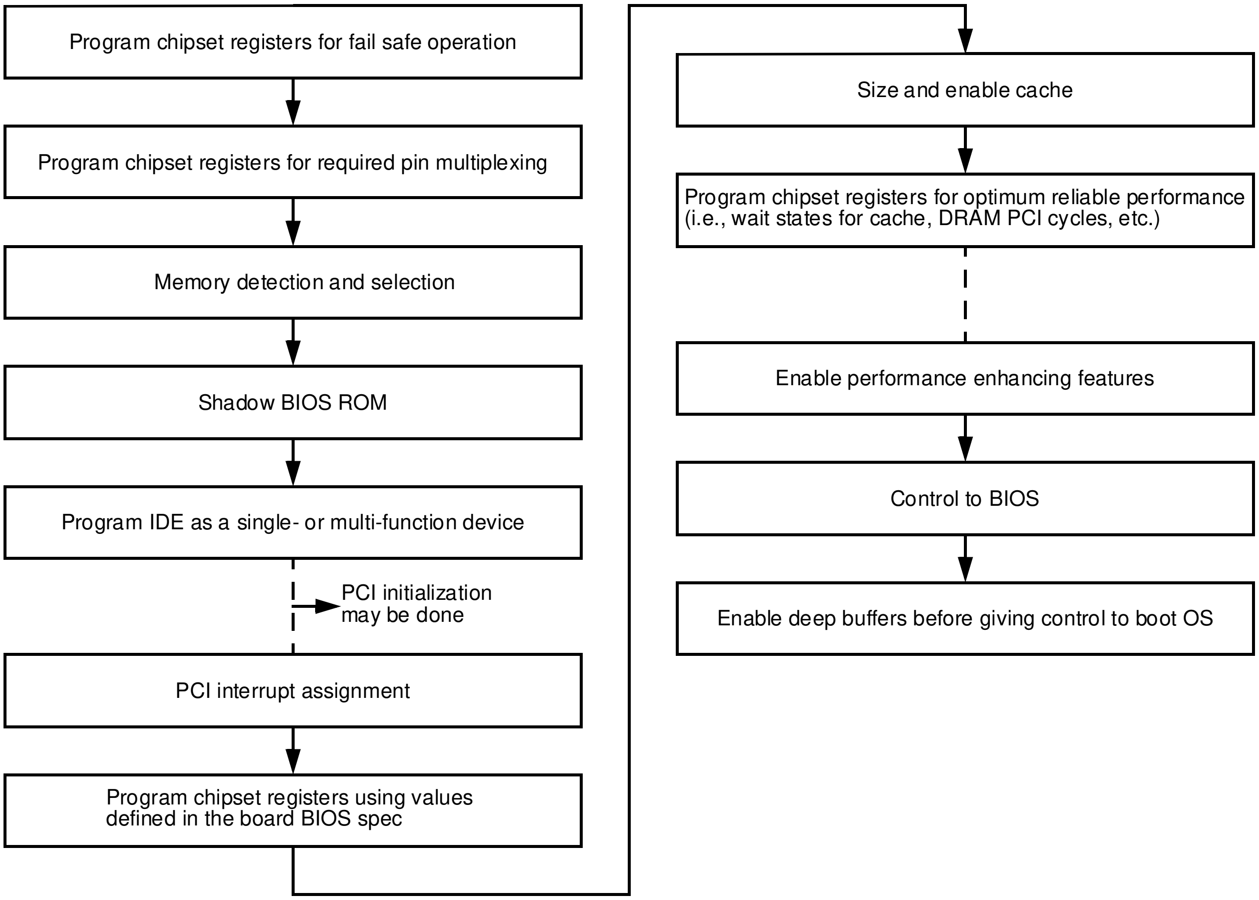
Discussion

BIOS Settings

Given below is a chipset specific illustrative flow chart that a BIOS needs to follow to program the chipset registers.

Tables 1 through 4 provide the register settings required of the chipset that the BIOS should ensure in order for the system to boot. These are fail safe boot values.

Figure 1 Boot Setting


Program chipset registers for fail safe operation
Program chipset registers for required pin multiplexing
Memory detection and selection
Shadow BIOS ROM
Program IDE as a single- or multi-function device
PCI initialization may be done
PCI interrupt assignment
Program chipset registers using values defined in the board BIOS spec
Size and enable cache
Program chipset registers for optimum reliable performance (i.e., wait states for cache, DRAM PCI cycles, etc.)
Enable performance enhancing features
Control to BIOS
Enable deep buffers before giving control to boot OS

OPTi Inc. makes no representations or warranties with respect to the design and documentation herein described and especially disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, OPTi Inc. reserves the right to revise the design and associated documentation and to make changes from time to time in the content without obligation of OPTi Inc. to notify any person of such revisions or changes.

Table 1 82C579 Fail Safe Boot Values

Loc.	Value
PCIDV0 PCI Config Register Space	
00h	45h
01h	10h
02h	69h
03h	C5h
04h	07h
05h	00h
06h	80h
07h	02h
08h	10h
09h	00h
0Ah	00h
0Bh	06h
0Ch	00h
0Dh	00h
0Eh	00h
0Fh	00h
10h	00h
11h	00h
12h	00h
13h	00h
14h	00h
15h	00h
16h	00h
17h	00h
18h	00h
19h	00h
1Ah	00h
1Bh	00h
1Ch	00h
1Dh	00h
1Eh	00h
1Fh	00h
20h	00h
21h	00h
22h	00h
23h	00h
24h	00h
25h	00h
26h	00h
27h	00h
28h	00h
29h	00h
2Ah	00h
2Bh	00h
2Ch	00h
2Dh	00h
2Eh	00h
2Fh	00h
30h	00h
31h	00h
32h	00h
33h	00h
34h	00h
35h	00h
36h	00h
37h	00h
38h	00h
39h	00h
3Ah	00h
3Bh	00h
3Ch	00h
3Dh	00h
3Eh	00h
3Fh	00h
40h	00h
41h	00h
42h	00h
43h	20h
44h[1]	01h
45h	00h
46h	00h
47h	00h
48h	00h
49h	00h
4Ah	00h
4Bh	00h
4Ch	40h[2]
4Dh	00h
4Eh	00h
4Fh	00h
SYSCFG Sys Cntrl Register Space	
00h	00h
01h	00h
02h	00
03h	00h
04h	00h
05h	00h
06h	00h
07h	00h
08h	00h
09h	00h
0Ah	00h
0Bh	00h
0Ch	00h
0Dh	00h
0Eh	00h
0Fh	00h
10h	00h
11h	00h
12h	00h
13h	80h
14h[1]	80h
15h	00h
16h	00h
17h	00h
18h	00h
19h	88h
1Ah	00h
1Bh	00h
1Ch	00h
1Dh	00h
1Eh	08h
1Fh	02h
20h	8Ah
21h	01h
22h	00h
23h	00h
24h	00h
25h	00h
26h	00h
27h	06h
28h	00h
29h	00h
2Ah	01h
2Bh	00h
2Ch	00h
2Dh	00h
2Eh	00h
2Fh	00h

[1] SYSCFG 14h must be written first followed by PCIDV0 44h[1]. It is important that PCIDV0 44h-47h is always a 32-bit write and in a tight sequence code.

[2] If L2 cache is being used in the system, 4Eh[6] should be set to 1.

Table 2 82C578 Fail Safe Boot Values

Loc.	Value
PCIDV1 PCI Config Register Space	
00h	45h
01h	10h
02h	68h
03h	C5h
04h	07h
05h	00h
06h	80h
07h	02h
08h	10h
09h	00h
0Ah	01h
0Bh	06h
0Ch	00
0Dh	00
0Eh	00h
0Fh	00h
10h	00h
11h	00h
12h	00h
13h	00h
14h	00h
15h	00h
16h	00h
17h	00h
18h	00h
19h	00h
1Ah	00h
1Bh	00h
1Ch	00h
1Dh	00h
1Eh	00h
1Fh	00h
20h	00h
21h	00h
22h	00h
23h	00h
24h	00h
25h	00h
26h	00h
27h	00h
28h	00h
29h	00h
2Ah	00h
2Bh	00h
2Ch	00h
2Dh	00h
2Eh	00h
2Fh	00h
30h	00h
31h	00h
32h	00h
33h	00h
34h	00h
35h	00h
36h	00h
37h	00h
38h	00h
39h	00h
3Ah	00h
3Bh	00h
3Ch	00h
3Dh	00h
3Eh	00h
3Fh	00h
40h	00h
41h	00h
42h	00h
43h	00h
44h	00h
45h	00h
46h	06h
47h	00h
48h	00h
49h	00h
4Ah	00h
4Bh	00h
4Ch	00h
4Dh	00h
4Eh	00h
4Fh	00h
50h	00h
51h	00h
52h	00h
53h	00h
54h	00h
55h	02h
56h	00h
57h	00h
58h	00h
59h	00h
5Ah	00h
5Bh	00h
5Ch	00h
5Dh	00h
5Eh	00h
5Fh	C0h
60h	00h
61h	00h
62h	00h
63h-FCh	00h
FDh	xxh
FEh	xxh
FFh	xxh
SYSCFG Pwr Mgmt Registers	
E0h	00h
E1h	00h
E2h	00h
E3h	00h
E4h	00h
E5h	00h
E6h	00h
E7h	00h
E8h	00h
E9h	00h
EAh	01h
EBh	01h
ECh	01h
EDh	00h
EEh	00h
EFh	00h
F0h	00h
F1h	00h
F2h	00h
F3h	00h
F4h	00h
F5h	00h
F6h	00h
F7h	00h
F8h	00h
F9h	00h
FAh	00h
FBh	00h
FCh	00h
FDh	00h
FEh	00h
FFh	10h

Table 3 Register Boot Values without Deep Buffers

Loc.	Value
SYSCFG **Sys Cntrl Register Space**	
00h	00h
01h	DDh
02h	03h
03h	EEh
04h	05h
05h	00h
06h	15h
07h	xxh
08h	6Ah
09h	00h
0Ah	00h
0Bh	00h
0Ch	40h
0Dh	02h
0Eh	43h
0Fh	24h
10h	83h
11h	08h
12h	00h
13h	83h
14h	80h
15h	A1h
16h	A4h
17h	02h
18h	50h
19h	00h
1Ah	00h
1Bh	00h
1Ch	00h
1Dh	20h
1Eh	CCh
1Fh	02h
20h	8Fh
21h	03h
22h	00h
23h	06h
24h	00h
25h	00h
26h	00h
27h	00h
28h	00h
29h	00h
2Ah	01h
2Bh	00h
2Ch	00h
2Dh	00h
2Eh	40h
2Fh	00h
PCIDV0 **PCI Config Register Space**	
80000000h	C5691045h
80000004h	02800007h
80000008h	06000010h
8000000Ch	00000000h
80000010h	00000000h
80000014h	00000000h
80000018h	00000000h
8000001Ch	00000000h
80000020h	00000000h
80000024h	00000000h
80000028h	00000000h
8000002Ch	00000000h
80000030h	00000000h
80000034h	00000000h
80000038h	00000000h
8000003Ch	00000000h
80000040h	20000000h
80000044h	00000007h
80000048h	00000000h
8000004Ch	00000000h
80000050h	00000000h
80000054h	00000000h
80000058h	00000000h
8000005Ch	00000000h
80000060h	00000000h
80000064h	00000000h
80000068h	00000000h
8000006Ch	00000000h
80000070h	00000000h
80000074h	00000000h
80000078h	00000000h
8000007Ch	00000000h
PCIDV1 **PCI Config Register Space**	
80000800h	C5681045h
80000804h	92800107h
80000808h	06010010h
8000080ch	00800000h
80000810h	00000000h
80000814h	00000000h
80000818h	00000000h
8000081Ch	00000000h
80000820h	00000000h
80000824h	00000000h
80000828h	00000000h
8000082Ch	00000000h
80000830h	00000000h
80000834h	00000000h
80000838h	00000000h
8000083Ch	00000000h
80000840h	20000000h
80000844h	40060001h
80000848h	C0000000h
8000084Ch	40000000h
80000850h	0004C000h
80000854h	000000D0h
80000858h	00000000h
8000085Ch	00000000h
80000860h	00000000h
80000864h	00000000h
80000868h	00000000h
8000086Ch	00000000h
80000870h	00000000h
80000874h	00000000h
80000878h	00000000h
8000087Ch	00000000h

Table 4 Register Boot Values with Deep Buffers

Loc.	Value
SYSCFG **Sys Cntrl Register Space**	
00h	00h
01h	DDh
02h	03h
03h	EEh
04h	05h
05h	00h
06h	15h
07h	xxh
08h	6Ah
09h	00h
0Ah	00h
0Bh	00h
0Ch	40h
0Dh	02h
0Eh	43h
0Fh	24h
10h	83h
11h	08h
12h	00h
13h	83h
14h	80h
15h	A1h
16h	A4h
17h	02h
18h	50h
19h	00h
1Ah	00h
1Bh	00h
1Ch	00h
1Dh	20h
1Eh	CCh
1Fh	02h
20h	8Fh
21h	03h
22h	00h
23h	06h
24h	00h
25h	00h
26h	00h
27h	00h
28h	00h
29h	00h
2Ah	0Dh
2Bh	00h
2Ch	21h
2Dh	00h
2Eh	48h
2Fh	00h
PCIDV0 **PCI Config Register Space**	
80000000h	C5691045h
80000004h	02800007h
80000008h	06000010h
8000000Ch	00000000h
80000010h	00000000h
80000014h	00000000h
80000018h	00000000h
8000001Ch	00000000h
80000020h	00000000h
80000024h	00000000h
80000028h	00000000h
8000002Ch	00000000h
80000030h	00000000h
80000034h	00000000h
80000038h	00000000h
8000003Ch	00000000h
80000040h	20000000h
80000044h	20003071h
80000048h	00000000h
8000004Ch	00000000h
80000050h	00000000h
80000054h	00000000h
80000058h	00000000h
8000005Ch	00000000h
80000060h	00000000h
80000064h	00000000h
80000068h	00000000h
8000006Ch	00000000h
80000070h	00000000h
80000074h	00000000h
80000078h	00000000h
8000007Ch	00000000h
PCIDV1 **PCI Config Register Space**	
80000800h	C5681045h
80000804h	92800107h
80000808h	06010010h
8000080Ch	00800000h
80000810h	00000000h
80000814h	00000000h
80000818h	00000000h
8000081Ch	00000000h
80000820h	00000000h
80000824h	00000000h
80000828h	00000000h
8000082Ch	00000000h
80000830h	00000000h
80000834h	00000000h
80000838h	00000000h
8000083Ch	00000000h
80000840h	20000000h
80000844h	40060001h
80000848h	C0000000h
8000084Ch	40000000h
80000850h	0004C000h
80000854h	000000D0h
80000858h	00000000h
8000085Ch	00000000h
80000860h	00000000h
80000864h	00000000h
80000868h	00000000h
8000086Ch	00000000h
80000870h	00000000h
80000874h	00000000h
80000878h	00000000h
8000087Ch	00000000h

Pin Muxing in the Viper Xpress+ Chipset

The Viper Xpress+ Chipset allows a great amount of flexibility to enable the system designer to tailor the design optimally to meet requirements. The BIOS programmer should obtain the hardware muxing information from the board designer to program the pin muxing registers. These values should be programmed once and should not be changed during warm resets.

Globally, the designer can program two bits in the 82C578 to enable a group of pins to take on different functions or can individually program bits to assign the desired functionality on a pin-by-pin basis. Table 5 shows the group-wise pin programming option and Table 6 shows the register bits used in pin programming (pin- and group-wise).

Table 5 82C578 Group-Wise Register Programmable Pins

	PCIDV1 44h[1:0]			
Pin No.	**00**	**01**	**10**	**11**
141	PIRQ2#	PIRQ2#	GPCS0##	PIRQ2#
143	PIRQ3#	PIRQ3#	EPMI2#	PIRQ3#
108	DACK0#	DACK0#	EDACK0	EDACK0
109	DACK1#	DACK1#	EDACK1	EDACK1
110	DACK2#	DACK2#	EDACKEN#	EDACKEN#
111	DACK3#	DACK3#	EDACK2	EDACK2

Note: The group-wise pin functionality of these pins will always be overridden by the pin-wise programmability.

Table 6 82C578 Pin Functionality Programming Register Bits

7	6	5	4	3	2	1	0
PCIDV1 41h			**Keyboard Control Register - Byte 1**				**Default = 00h**
			Keyboard emulation: 0 = Enable - Pin 12 functions as A20M# output 1 = Disable - Pin 12 functions as KBRST input				
PCIDV1 42h			**Interrupt Edge/Level Control Register - Byte 0**				**Default = 00h**
							Pin 122 functionality: 0 = DREQ6 1 = EPMI0#

Table 6 82C578 Pin Functionality Programming Register Bits (cont.)

7	6	5	4	3	2	1	0
PCIDV1 44h			**Pin Functionality Register 1 - Byte 0**				**Default = 00h**
Pin 111 functionality:[1] 0X = Controlled by bits [1:0] 10 = DACK7# 11 = Reserved If set to 10, the setting on bits [1:0] will not affect the functionality that this pin takes on.		Pin 109 functionality:[2] 0X = Controlled by bits [1:0] 10 = DACK6# 11 = Reserved If set to 10, the setting on bits [1:0] will not affect the functionality that this pin takes on.		Pin 108 functionality: 0X = Controlled by bits [1:0] 10 = DACK5# 11 = Reserved		DACK/PIRQ[3:2]# group-wise programmable pin functionalities: 00 = Explicit DACK[3:0]#, PIRQ[3:2]# 01 = Explicit DACK[7:5,3,1,0]#, GPCS0#, PIRQ[3:2]# 10 = Encoded EDACK[2:0], EDACKEN# 11 = Encoded EDACK[2:0], EDACKEN#, PIRQ[3:2]# Pin-wise, these functions may be overridden by GPCS[x]#, EPMI[x]#, and DACK[7:5]# (for DACK3#, DACK1#, and DACK0#).	
(1) Pin 111 can take on the following functionalities - DACK3#, EDACK2, or DACK7#. DACK3# and EDACK2 are group-wise programmable, and both are pin-wise programmable with DACK7#. (2) Pin 109 can take on the following functionalities - DACK1#, EDACK1, or DACK6#. DACK1# and EDACK1 are group-wise programmable, and both of them are pin-wise programmable with DACK6#.							
PCIDV1 45h			**Pin Functionality Register 1 - Byte 1**				**Default = 00h**
		Pin 143 functionality:[1] If PCIDV1 51h[4] = 0: 0 = Controlled by PCIDV1 44h[1:0] 1 = EPMI2# (If this bit is set and 51h[4] = 0, pin 143 takes on the EPMI2# functionality regardless of the setting of 44h[1:0])	Pin 141 functionality: This bit determines the group-wise functionality of the PIRQ2#+ GPCS0#. These two functionalities are group-wise programmable. 0 = Controlled by PCIDV1 44h[1:0] and PCIDV1 51h[3] = 0 1 = Reserved	Pin 140 functionality: 00 = PIRQ1# 01 = IRQ0 1X = Reserved		Pin 139 functionality: 00 = PIRQ0# 01 = EPMI1# 1X = Reserved	
(1) This pin can take on any of the these functionalities - PIRQ3#, GPCS1#, or EPMI2#. PIRQ3# and GPCS1# functionalities are group-wise programmable, and those two functionalities are pin-wise programmable with EPMI2#.							
PCIDV1 48h			**Pin Functionality Register 2 - Byte 0**				**Default = 00h**
Pin 120 functionality: 00 = DREQ3 01 = DREQ3/7 10 = DREQ7 11 = Reserved		Pin 117 functionality: 00 = DREQ1 01 = DREQ1/6 10 = DREQ6 11 = Reserved		Pin 116 functionality: 00 = DREQ0 01 = DREQ0/5 10 = DREQ5 11 = Reserved			
PCIDV1 49h			**Pin Functionality Register 2 - Byte 1**				**Default = 00h**

Table 6 82C578 Pin Functionality Programming Register Bits (cont.)

7	6	5	4	3	2	1	0
Pin 136 functionality: 0 = IRQ15 1 = Reserved	Pin 134 functionality: 0X = IRQ12 10 = MPIRQ2#/3# 11 = Reserved		Pin 132 functionality: 0 = IRQ10 1 = MIRQ10/12	Pin 128 functionality: 0 = IRQ6 1 = MPIRQ0#/ 1# Also see PCIDV1 53h[7].	Pin 126 functionality: 0 = IRQ4 1 = MIRQ4/6	Pin 123 functionality: 00 = DREQ7 01 = EPMI3# 1X = Reserved	
PCIDV1 4Fh	**Miscellaneous Control Register - Byte 1**						**Default = 00h**
Pin 112 functionality: 0 = DACK5# (also see PCIDV1 44h[1:0]) 1 = PPWRL# +PPWRL2						Pin 113 functionality: 0 = Controlled by PCIDV1 44h[1:0] 1 = GPCS2# If set to 1, the PCIDV1 44h[1:0] setting will not affect the functionality that this pin takes on.	
PCIDV1 51h	**Interrupt Trigger Control Register - Byte 1**						**Default = 00h**
Pin 104 functionality: 0 = 32KHz 1 = PREQ3# Also see PCIDV1 5Eh[6].	Pin 90 functionality: 0 = ZEROWS# 1 = PGNT3#	Pin 110 functionality: 0 = DACK2# 1 = GPCS2# Also see PCIDV1 44h[1:0].	Pin 143 functionality: 0 = PIRQ3# 1 = GPCS1# Also see PCIDV1 45h[5].	Pin 141 functionality: 0 = PIRQ2# 1 = GPCS0#			
PCIDV1 53h	**Interrupt Multiplexing Control Register - Byte 1**						**Default = 00h**
Pin functionality: 0 = Pin #: 125 = IRQ3 127 = IRQ5 128 = IRQ6 129 = IRQ7 131 = IRQ9 133 = IRQ11 1 = Pin #: 125 = MIRQ3/5 127 = MIRQ7/9 128 = MIRQ11/ 15 129 = EPMI1# 131 = EPMI2# 133 = GMIRQ	Pin 146 functionality: 0 = PREQ2# 1 = EPMI0#						
PCIDV1 55h	**PCI Master Control Register - Byte 1**						**Default = 00h**

Table 6 82C578 Pin Functionality Programming Register Bits (cont.)

7	6	5	4	3	2	1	0
			SERIRQ# muxing on Pin 1: 0 = Disable 1 = Enable(1)				
(1) Also these PCIDV1 register bits must be set: 54h[4] = 1, 59h[3] = 0, and 5Fh[4] = 0.							
PCIDV1 59h			**Pin Functionality Register 3**				**Default = 00h**
				EPMI0# muxing on Pin 1: 0 = Disable 1 = Enable(1)			Pin 106 functionality: 0 = RTCRD# 1 = PGNT3# For SDACK2# function on this pin, see PCIDV1 5Eh[6]. For PCI soft reset generation through RTCRD#, see PCIDV1 61h[7] and 62h[7].
(1) Also these PCIDV1 register bits must be set: 54h[4] = 1, 55h[4] = 0, and 5Fh[4] = 0.							
PCIDV1 5Eh			**Steerable DRQ Control Register**				**Default = 00h**
SDRQ/ SDACK# functions on pins 105 and 107: 0 = Disable 1 = Enable(1)	SDRQ/ SDACK# functions on pins 104 and 106: 0 = Disable 1 = Enable(2)						
(1) Pin 105 functions as SDRQ1 and pin 107 functions as SDACK1#. Also refer to PCIDV1 60h[4]. (2) Pin 104 functions as SDRQ2 and pin 106 functions as SDACK2#. Also refer to PCIDV1 51h[7] and 59h[0].							
PCIDV1 5Fh			**Steerable IRQ Control Register**				**Default = 00h**
Pin 52 functionality: 0 = Reserved 1 = MSGN2S	Pin 154 functionality: 00 = Reserved 01 = Reserved 10 = MSGS2N 11 = USBGNT#		SERIRQ# muxing on Pin 1: 0 = Disable 1 = Enable(1)				
(1) Also these PCIDV1 register bits must be set: 54h[4] = 1, 55h[4] = 0, and 59h[3] = 0.							
PCIDV1 60h			**USB Interrupt Control Register**				**Default = 00h**

Table 6 82C578 Pin Functionality Programming Register Bits (cont.)

7	6	5	4	3	2	1	0
			Pins 105 and 107 functionality: 0 = RTCAS+ SDRQ1 on pin 105 and RTCWR#+ SDACK1# on pin 107 1 = PREQ4# on pin 105 and PGNT4# on pin 107				
PCIDV1 61h			**PCI Reset Control Register**				**Default = 00h**
PCI soft reset generation through RTCRD#: 0 = No action 1 = Generate a 100µs PCI reset pulse if PCIDV1 62h[7] = 0				USBGNT# thru RTCWR# (pin 107): 0 = Enabled only if PCIDV1 5Fh[7] = 1 1 = Disable (no USBGNT# functionality)	Buffered DMA LOCK# thru SERIRQ#/RTCAS pins: 00 = No BFLOCK# thru either pin 01 = BFLOCK# thru SERIRQ# pin (pin 1) 10 = BFLOCK# thru RTCAS pin (pin 105) 11 = Reserved These bit settings will override PCIDV1 60h[4], 5Fh[4], 59h[3] and 5Eh[7].		
PCIDV1 62h			**Emulation Control Register**				**Default = 00h**
Pin functionality: 0 = PCIRES# enable 1 = PCIRES# disable							

Table 7 82C579 Pin Functionality Programming Register Bits

7	6	5	4	3	2	1	0
SYSCFG 26h			**ISA Control Register**				**Default = 00h**
					USBGNT#: 0 = Through messaging protocol 1 = Through pin 58		
SYSCFG 2Dh			**Bank-wise EDO Timing Selection Register**				**Default = 00h**
DIRTYI pin (pin 158) mux: 0 = MREF# (MCACHE) 1 = NVMCS							
SYSCFG 2Eh			**PCI Master Register**				**Default = 00h**
BFLOCK pin (pin 177) control: 0 = Disable 1 = Enable	Pin 102 functionality: 0 = USBCLK 1 = REFRESH#	MSGN2S/ MSGS2N bus enabling: 0 = Disable 1 = Enable					

DRAM Subsystem

After a power-on reset when the BIOS attempts to configure the DRAM subsystem, the registers that affect the DRAM subsystem are located in the 82C579. All the registers that control the DRAM subsystem are accessed by the 22h, 24h indexing scheme (to access the System Control Register Space - SYSCFG).

The Viper Xpress+ Chipset supports up to six banks of DRAM. Given below is a step-by-step procedure for initializing the DRAM subsystem of the chipset. Table 8 shows the registers associated with configuration of the DRAM subsystem.

Step 1

Program SYSCFG 13h[7] = 1. This will enable the Viper Xpress+ Chipset to fully decode the incoming address.

Step 2

The BIOS should then program the size of each DRAM bank to be the maximum size before it determines the exact size of memory in each bank.

SYSCFG	Logical Bank Addressed
13h[2:0]	Logical Bank 0
13h[6:4]	Logical Bank 1
14h[2:0]	Logical Bank 2
14h[6:4]	Logical Bank 3
19h[2:0]	Logical Bank 4
19h[6:4]	Logical Bank 5

Notes:

A. SYSCFG 19h[7] and 19h[3] should be set to 1 before starting DRAM sizing.

B. The maximum DRAM size setting in each register is 16Mx72 which corresponds to a 3-bit binary code of “111”.

E. Above all, the L1 and L2 caches must be disabled when the DRAM subsystem is being initialized.

Step 3

Size Logical Banks 0 through 5 and program the appropriate 3-bit binary code in the appropriate registers.

Table 8 DRAM Configuration Registers

7	6	5	4	3	2	1	0
SYSCFG 13h	**Memory Decode Control Register 1**						**Default = 00h**
Memory decode select: This bit must be set to 1 for full decode (maximum flexibility in choosing different DRAM configurations)	Full decode for logical Bank 1 (RAS1#) if SYSCFG 13h[7] is set: 000 = 0Kx72 100 = 2Mx72 001 = 256Kx72 101 = 4Mx72 010 = 512Kx72 110 = 8Mx72 011 = 1Mx72 111 = 16Mx72			SMRAM: 0 = Disable 1 = Enable	Full decode for logical Bank 0 (RAS0#) if SYSCFG 13h[7] is set: 000 = 0Kx72 100 = 2Mx72 001 = 256Kx72 101 = 4Mx72 010 = 512Kx72 110 = 8Mx72 011 = 1Mx72 111 = 16Mx72		
SYSCFG 14h	**Memory Decode Control Register 2**						**Default = 00h**
82C576 mode: 0 = Normal mode 1 = Clocked mode (Must = 1 for EDO timing)	Full decode for logical Bank 3 (RAS3#) if SYSCFG 13h[7] is set: 000 = 0Kx72 100 = 2Mx72 001 = 256Kx72 101 = 4Mx72 010 = 512Kx72 110 = 8Mx72 011 = 1Mx72 111 = 16Mx72			SMRAM control: **Inactive SMIACT#:** 0 = Disable SMRAM 1 = Enable SMRAM(1) **Active SMIACT#:** 0 = Enable SMRAM for both Code and Data(1) 1 = Enable SMRAM for Code only (1)	Full decode for logical Bank 2 (RAS2#) if SYSCFG 13h[7] is set: 000 = 0Kx72 100 = 2Mx72 001 = 256Kx72 101 = 4Mx72 010 = 512Kx72 110 = 8Mx72 011 = 1Mx72 111 = 16Mx72		
(1) If SYSCFG 13h[3] is set.							
SYSCFG 19h	**Memory Decode Control Register 3**						**Default = 00h**
Reserved: Must be written to 1.	Full decode for logical bank 5 (RAS5#) if SYSCFG 13h[7] is set: 000 = 0Kx72 100 = 2Mx72 001 = 256Kx72 101 = 4Mx72 010 = 512Kx72 110 = 8Mx72 011 = 1Mx72 111 = 16Mx72			Reserved: Must be written to 1.	Full decode for logical bank 4 (RAS4#) if SYSCFG 13h[7] is set: 000 = 0Kx72 100 = 2Mx72 001 = 256Kx72 101 = 4Mx72 010 = 512Kx72 110 = 8Mx72 011 = 1Mx72 111 = 16Mx72		

EDO DRAM Auto Detection

This section of the document explains the mechanism used to automatically detect EDO DRAM SIMMs located on the motherboard by the Viper Xpress+ Chipset.

Detection between WE# controlled EDO DRAMs and Fast Page Mode DRAMs

Differentiating between these two type of DRAMs is a little bit more complex because FP Mode DRAMs specified at 70ns, could perform better than their rating and thus may even work with tighter timing. A second factor that makes the detection complex is the large capacitive load presented by the DRAM on the MD bus. This capacitive load manifests itself in the form of large discharge times, thereby retaining the last driven value on the bus for long periods of time.

A solution to this problem is to latch data into the chipset after a significantly large time after the CAS has been pulled high. Also an attempt could be made to generate a conflict on the bus, thereby discharging the bus and then attempting to read back the data present on the MD bus.

When SYSCFG 1Fh[6] (in the 82C579) is set to “1” it puts the Viper Xpress+ Chipset in the Mode of EDO DRAM detection. This will cause a quad-word to be read in 4µs. When bit 6 is set, setting bit 7 to a “1” will cause a conflict to be generated at about 2µs, if necessary.

An algorithm is given below to use this feature and detect the type of DRAM used on the board. Refer to

Size the first bank

1. Set SYSCFG 1Fh[6] = 1.
2. Set SYSCFG 1Fh[7] = 1. This step needs to be done only if the user desires to create a conflict on the bus.
3. Write a known pattern to a pre-determined location in DRAM.
4. Write a second different pattern to a second pre-determined location in the DRAM.
5. Read back data from the first pre-determined location in DRAM.
6. Read back data from second pre-determined location.
7. If data read back is the same as the data written to the first pre-determined location, then the DRAM SIMMs are WE# EDO SIMMs, otherwise they are Fast Page Mode DRAM SIMMs.
8. Set SYSCFG 1Ch[2] = 1 if Bank 0 = EDO
   Set SYSCFG 1Ch[2] = 0 if Bank 0 = FP DRAM
   ....
   ....
   ....
   Set SYSCFG 1Ch[7] = 1 if Bank 5 = EDO
   Set SYSCFG 1Ch[7] = 0 if Bank 5 = FP DRAM
9. Repeat for all banks.

Note: While EDO auto detection is in progress, hidden refresh should be disabled.

Table 9 Register Bits Associated with EDO DRAM Auto Detection

7	6	5	4	3	2	1	0
SYSCFG 1Ch			**EDO DRAM Control Register**				**Default = 00h**
EDO DRAM usage: Each bit is set to a 1 if the user is using EDO DRAMs in each of the available six banks. Bit 2 corresponds to Bank 0 and bit 7 corresponds to Bank 5, yielding a total of six banks that the user can populate. 0 = Standard page mode DRAM 1 = EDO DRAM							
SYSCFG 1Fh			**EDO Timing Control Register**				**Default = 00h**
0 = Normal 1 = Generate conflict during EDO detection (bit 6 set) if necessary	0 = Normal (fast page mode) 1 = Detect EDO						

Shadowing the ROM Area

The BIOS needs to shadow the F0000 area so that the BIOS code can execute out of local memory. Given below is an algorithm to achieve this and shows the 82C578 and 82C579 associated register bits.

1. The BIOS should set PCIDV1 4Bh[7:6] = 00. This enables generation of the ROMCS# by the 82C578 whenever the address is in the F0000 segment.
2. The BIOS should then set SYSCFG 06h[3:2] = 10. This causes all reads to the F0000 segment to go across the PCI bus while all writes will be performed on the DRAM.
4. The BIOS should then copy the contents of the ROM in the F0000 segment to DRAM.
5. The BIOS should set PCIDV1 4Bh[7:6] = 11. This disables generation of the ROMCS# by the 82C578 whenever the address is in the F0000 segment.
6. The BIOS should then set register SYSCFG 06h[3:2] = 11. This causes all reads and writes to be performed on the DRAM.

Table 10 Register Bits Associated with ROM Shadowing

7	6	5	4	3	2	1	0
PCIDV1 4Bh			**ROMCS# Range Control Register - Byte 1**				**Default = 00h**
ROMCS# for FFFF8000h-FFFFFFFFh segment: 0 = Enable 1 = Disable	ROMCS# for FFFF0000h-FFFF7FFFh segment: 0 = Enable 1 = Disable	ROMCS# for FFFE8000h-FFFEFFFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFE0000h-FFFE7FFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFD8000h-FFFDFFFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFD0000h-FFFD7FFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFC8000h-FFFCFFFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFC0000h-FFFC7FFFh segment: 0 = Disable 1 = Enable
SYSCFG 06h			**Shadow RAM Control Register 3**				**Default = 00h**
DRAM hole in system memory from 80000h-9FFFFh:[1] 0 = No hole in memory 1 = Enable hole in memory	Wait state addition for PCI master snooping: 0 = Do not add a wait state for the cycle access finish to do the snooping 1 = Add a wait state for the cycle access to finish and then do the snooping	C0000h-C7FFFh cacheability: 0 = Not Cacheable 1 = Cacheable in L1 and L2 (L1 disabled by SYSCFG 08h[0])	F0000h-FFFFFh cacheability: 0 = Not Cacheable 1 = Cacheable in L1 and L2 (L1 disabled by SYSCFG 08h[0])	F0000h-FFFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM If SYSCFG 04h[2] = 1, then the E0000h-EFFFFh read/write control should have the same setting as this.		E0000h-EFFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM	
(1) This setting gives the user the option to have some other device in the address range 80000h-9FFFFh instead of system memory. When bit 7 is set, the 82C579 will not start the system DRAM controller for accesses to this particular address range.							

PCI Bus Master IDE Configuration Requirements

The system IDE controller configuration is done by the BIOS in two phases. The first phase takes place when the devices on the PCI bus are initialized. During this phase, the BIOS assigns interrupts and sets the level of the interrupts. The second phase of configuring the IDE controller takes place when the features in the system are enabled. During this phase, the IDE controller is configured for optimum performance based on the drives installed in the system. This is when the timing of the IDE controller is programmed into the internal registers of the controller.

The following section includes the tasks that are done by the BIOS during both of these phases. The BIOS is required to separate the two phases based on the guidelines provided above.

In order to optimize the PCI IDE Module for both DMA and PIO operations, the following steps are expected to be fulfilled by the system BIOS:

1. Enable the PCI IDE Module: PCIDV1 4Fh[6] = 1.
2. Configure the PCI IDE Module through Mechanism #1 as Bus #0, Device #1, and Function #1. Also set SYSCFG FFh[4] = 1 (82C578).
3. Detect the PCI bus frequency and record it into IDE I/O Address 1F5h[0]. For a synchronous PCI Viper Xpress+ system, the PCI bus frequency equals the system bus frequency divided by two.

   If a 25MHz PCI bus is detected, a 1 should be written into the IDE I/O Address 1F5h[0] and 01 should be written into PCIDV1 47h[5:4] for the proper ISA clock divisor as well.

   If the Viper Xpress+ system is configured for an asynchronous PCI clocking scheme, there is no need to perform this checking since an external 33MHz (power-up default) is feeding into the IDE module.
4. The default IDE ownership at PCIIDE 40h[4] should be set to 1. This allows the multiplexed ISA/IDE bus always park to the IDE Module.
5. The 32-byte read prefetch FIFO should be enabled by setting PCIIDE 40h[5] = 1.
6. Concurrent refresh and IDE cycles should be enabled by setting PCIDV1 52h[0] = 1.
7. PCI IDE one wait state reads for primary and secondary channels should be enabled at IDE I/O Address 1F3h[4].
8. Read prefetch for primary and secondary channels should be enabled at IDE I/O Address 1F6h[6] and 176h[6] accordingly.
9. Enable master capability at PCIIDE 04h[2]. Once mastering is enabled, set PCIDV1 54h[7:4] = Fh.
10. Assign value for bus master IDE base address at PCIIDE 20h-23h.
11. Depending on the capabilities of the system's hard drives and the PCI bus frequency, setup the IDE timings accordingly. The applicable SET_FEATURES commands (i.e., Flow Control Enable) should be issued to the corresponding IDE drives as well.
12. If no device is in the primary slave (Drive 1) location, set the command pulse and recovery time (1F0h/1F1h, Index-1) to correspond to PIO Mode 0.
13. If no device is in the secondary master (Drive 0) location or slave (Drive 1) location, set the command pulse and recovery time (170h/171h, Index-0 and 170h/171h, Index-1) to correspond to PIO Mode 0.

Table 11 PCI Bus Master IDE Configuration Associated Register Bits

7	6	5	4	3	2	1	0
PCIDV1 4Fh			**Miscellaneous Control Register - Byte 1**				**Default = 00h**
	IDE functionality support: 0 = Disable 1 = Enable						
SYSCFG FFh			**General Purpose Chip Select Control Register**				**Default = 00h**
			Reserved: Must be written to 1.				

Table 11 PCI Bus Master IDE Configuration Associated Register Bits

7	6	5	4	3	2	1	0
I/O Address 1F5h			**Strap Register**				**Default = xxh**
							PCI CLK speed: 0 = 33MHz 1 = 25MHz
PCIDV1 47h			**Cycle Control Register 1 - Byte 1**				**Default = 00h**
		ATCLK frequency select: 00 = LCLK÷4 10 = LCLK÷2 01 = LCLK÷3 11 = LCLK					
PCIIDE 40h			**IDE Initialization Control Register**				**Default = 00h**
		Enhanced slave: 0 = 82C621A-compatible mode, uses a 16-byte FIFO in PIO Mode 1 = Enhanced mode, uses a 32-byte FIFO in PIO Mode	Reserved: Must be written to 1.				
PCIDV1 52h			**Interrupt Multiplexing Control Register - Byte 0**				**Default = 00h**
							Concurrent refresh and IDE cycle: 0 = Disable 1 = Enable ISA devices that rely on accurate refresh addresses for proper operation should disable this bit.
I/O Address 1F3h			**Control Register**				**Default = xxh**
			Enable one wait state read: 0 = 2 WS minimum 1 = 1 WS minimum for data reads				

Table 11 PCI Bus Master IDE Configuration Associated Register Bits

7	6	5	4	3	2	1	0
I/O Address 1F6h			**Miscellaneous Register**				**Default = xxh**
	Read prefetch: 0 = Disable 1 = Enable						
I/O Address 176h			**Miscellaneous Register**				**Default = xxh**
	Read prefetch: 0 = Disable 1 = Enable						
PCIIDE 04h			**Command Register - Byte 0**				**Default = 4xh**
					IDE controller becomes a PCI master to generate PCI accesses: 0 = Disable 1 = Enable **Note:** This bit must be explicitly programmed.		
PCIDV1 54h			**PCI Master Control Register - Byte 0**				**Default = 00h**
PCI master write X-1-1-1: 0 = Disable 1 = Enable	PCI master read X-1-1-1: 0 = Disable 1 = Enable	PCI master/IDE concurrence: 0 = Disable 1 = Enable (Also see PCIIDE 42h[4] and [2])	New AHOLD protocol: 0 = Disable 1 = Enable (use HREQ to latch AHOLD)				
PCIIDE 20h-23h			**Bus Master IDE Base Address Register**				**Default = 01000080h**
This register is the I/O base address indicator for the Bus Master IDE Registers. The address block has a size of 16 bytes. Bits [31:0] correspond to: 20h = [7:0], 21h = [15:8], 22h = [23:16], 23h = [31:24]. - Bits [3:0] are read-only and default to 0001. - Bits [31:4] are writable.							
I/O Address 1F0h, Index-1			**Read Cycle Timing Register-B**[1]				**Default = xxh**
Read pulse width: The value programmed in this register plus one determines the DRD# pulse width in LCLKs (for a 16-bit read from the IDE Data Register).[2]				Read recovery time: The value programmed in this register plus two determines the recovery time between the end of DRD# and the next DA[2:0]/DCSx# being presented (after a 16-bit read from the IDE Data Register), measured in LCLKs.[2]			

[1] Read Cycle Timing Register-B shares the I/O address with Read Cycle Timing Register-A, indexed by 1F6h[0]. It controls the read cycle timing of the IDE Data Register for the drive not selected by 1F3h[3:2] if 1F3h[7] = 1.

[2] See Table 14 or Table 15 (of this document).

Table 11 PCI Bus Master IDE Configuration Associated Register Bits

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="3"><b>I/O Address 1F1h, Index-1</b></td><td colspan="3"><b>Write Cycle Timing Register-B[1]</b></td><td colspan="2"><b>Default = xxh</b></td></tr>
<tr><td colspan="4">Write pulse width:
The value programmed in this register plus one determines the DWR# pulse width in LCLKs (for a 16-bit write from the IDE Data Register).[2]</td><td colspan="4">Write recovery time:
The value programmed in this register plus two determines the recovery time between the end of DWR# and the next DA[2:0]/DCSx# being presented (after a 16-bit write from the IDE Data Register), measured in LCLKs.[2]</td></tr>
<tr><td colspan="8">(1) Write Cycle Timing Register-B shares the I/O address with Write Cycle Timing Register-A, indexed by 1F6h[0]. It controls the write cycle timing of the IDE Data Register for the drive not selected by 1F3h[3:2] if 1F3h[7] = 1.
(2) See Table 14 or Table 15 (of this document).</td></tr>
<tr><td colspan="8"></td></tr>
<tr><td colspan="3"><b>I/O Address 170h, Index-0</b></td><td colspan="3"><b>Read Cycle Timing Register-A[1]</b></td><td colspan="2"><b>Default = xxh</b></td></tr>
<tr><td colspan="4">Read pulse width:
The value programmed in this register plus one determines the DRD# pulse width in LCLKs (for a 16-bit read from the IDE Data Register).[2]</td><td colspan="4">Read recovery time:
The value programmed in this register plus two determines the recovery time between the end of DRD# and the next DA[2:0]/DCSx# being presented (after a 16-bit read from the IDE Data Register), measured in LCLKs.[2]</td></tr>
<tr><td colspan="8">(1) Read Cycle Timing Register-A shares the I/O address with Read Cycle Timing Register-B, indexed by 176h[0]. It controls the read cycle timing of the IDE Data Register for the drive selected by 173h[3:2].
(2) See Table 14 or Table 15 (of this document).</td></tr>
<tr><td colspan="8"></td></tr>
<tr><td colspan="3"><b>I/O Address 171h, Index-0</b></td><td colspan="3"><b>Write Cycle Timing Register-A[1]</b></td><td colspan="2"><b>Default = xxh</b></td></tr>
<tr><td colspan="4">Write pulse width:
The value programmed in this register plus one determines the DWR# pulse width in LCLKs (for a 16-bit write from the IDE Data Register).[2]</td><td colspan="4">Write recovery time:
The value programmed in this register plus two determines the recovery time between the end of DWR# and the next DA[2:0]/DCSx# being presented (after a 16-bit write from the IDE Data Register), measured in LCLKs.[2]</td></tr>
<tr><td colspan="8">(1) Write Cycle Timing Register-A shares the I/O address with Write Cycle Timing Register-B, indexed by 176h[0]. It controls the write cycle timing of the IDE Data Register for the drive selected by 173h[3:2].
(2) See Table 14 or Table 15 (of this document).</td></tr>
<tr><td colspan="8"></td></tr>
<tr><td colspan="3"><b>I/O Address 170h, Index-1</b></td><td colspan="3"><b>Read Cycle Timing Register-B[1]</b></td><td colspan="2"><b>Default = xxh</b></td></tr>
<tr><td colspan="4">Read pulse width:
The value programmed in this register plus one determines the DRD# pulse width in LCLKs (for a 16-bit read from the IDE Data Register).[2]</td><td colspan="4">Read recovery time:
The value programmed in this register plus two determines the recovery time between the end of DRD# and the next DA[2:0]/DCSx# being presented (after a 16-bit read from the IDE Data Register), measured in LCLKs.[2]</td></tr>
<tr><td colspan="8">(1) Read Cycle Timing Register-B shares the I/O address with Read Cycle Timing Register-A, indexed by 176h[0]. It controls the read cycle timing of the IDE Data Register for the drive not selected by 173h[3:2] if 173h[7] = 1.
(2) See Table 14 or Table 15 (of this document).</td></tr>
<tr><td colspan="8"></td></tr>
<tr><td colspan="3"><b>I/O Address 171h, Index-1</b></td><td colspan="3"><b>Write Cycle Timing Register-B[1]</b></td><td colspan="2"><b>Default = xxh</b></td></tr>
<tr><td colspan="4">Write pulse width:
The value programmed in this register plus one determines the DWR# pulse width in LCLKs (for a 16-bit write from the IDE Data Register).[2]</td><td colspan="4">Write recovery time:
The value programmed in this register plus two determines the recovery time between the end of DWR# and the next DA[2:0]/DCSx# being presented (after a 16-bit write from the IDE Data Register), measured in LCLKs.[2]</td></tr>
<tr><td colspan="8">(1) Write Cycle Timing Register-B shares the I/O address with Write Cycle Timing Register-A, indexed by 176h[0]. It controls the write cycle timing of the IDE Data Register for the drive not selected by 173h[3:2] if 173h[7] = 1.
(2) See Table 14 or Table 15 (of this document).</td></tr>
</table>

Programming PCI Bus Master IDE Timing

The power-up default of the IDE module is PIO Mode 0 for all four IDE devices. However, any of the IDE devices can be programmed up to PIO Mode 3. In addition, there are four sets of registers (two sets for each channel) that allow specific timings to be programmed on a “per-device” basis, see Table 12. These register sets are shared by the DMA and PIO operations. With the IDENTIFY_DRIVE command information, the BIOS can optimize the IDE timing for each drive individually by programming these registers.

Any combination of hard drives or ATAPI devices may be connected to the two channels of the IDE module in a four-drive configuration. For each channel, both sets of registers and the power-up default can be used to control any devices in the channel. See Table 13 for details.

Table 14 and Table 15 show the timing and recommended register settings for various IDE modes defined in the Enhanced IDE Specifications. They include PIO transfer, Single-Word DMA transfer, and Multi-Word DMA transfer modes. The actual cycle time equals the sum of actual command active time and actual command inactive (command recovery and address setup) time. These three timing requirements shall be met. In some cases, the minimum cycle time requirement is greater than the sum of the command pulse and command recovery time. This means either the command active (command pulse) time, or command inactive (command recovery and address setup) time can be lengthened to ensure that the minimum cycle times are met.

Figure 2 is a flow chart that describes how to program the primary channel of the MIDE interface. For the secondary channel, a similar procedure can be done by changing all the indexes from 1Fxh to 17xh.

Table 12 Registers for Programming IDE Timing

Name	Address	
Timing Registers-A for Primary Channel	1F0h/1F1h - Index-0	1F3h, 1F5h, and 1F6h are shared by both indexes
Timing Registers-B for Primary Channel	1F0h/1F1h - Index-1	
Timing Registers-A for Secondary Channel	170h/171h - Index-0	173h, 175h, and 176h are shared by both indexes
Timing Registers-B for Secondary Channel	170h/171h - Index-1	

Table 13 REGTIMx Programming Options

REGTIM0 1F3h[2]/173h[2]	REGTIM1 1F3h[3]/173h[3]	REGTIM2 1F3h[7]/173h[7]	Master/Drive 0 Controlled by:	Slave/Drive 1 Controlled by:
1(1)	0	1	Index-0	Index-1
0	1	1	Index-1	Index-0
0	0	1	Index-1	Index-1
1	0	0	Index-0	Default(2)
0	1	0	Default(2)	Index-0
0	0	0	Default(2)	Default(2)
1	1	X	Index-0	Index-0

(1) Recommended configuration

(2) Refer to PCIIDE 40h[1:0]

Table 14 16-Bit Timing Parameters with 33/30MHz PCI Bus

Parameter: Register Bits	Dimension	IDE Transfer Modes											
		PIO Modes						Multi-Word DMA Modes			Single-Word DMA Modes		
		0	1	2	3	4	5	0	1	2	0	1	2
Address Setup: 1F6h/176h[5:4]	Bit values in hex	2	1	1	1	0	0	0	0	0	0	0	0
	Timing in LCLKs(1)	3	2	2	2	1	1	1	1	1	1	1	1
	Enhanced IDE Spec in ns(2)	70	50	30	30	25	N/S	N/A	N/A	N/A	N/A	N/A	N/A
R/W Command Pulse: 1F0h/170h/1F1h/ 171h[7:4], Index-0/1	Bit values in hex	5	4	3	2	2	2	7	2	2	F	8	4
	Timing in LCLKs(1)	6	5	4	3	3	3	8	3	3	16	9	5
	Enhanced IDE Spec in ns(2)	165	125	100	80	70	N/S	215	80	70	480	240	120
R/W Recovery Time: 1F0h/170h/1F1h/ 171h[3:0], Index-0/1	Bit values in hex	9	4	0	0	0	0	6	0	0	D	4	0
	Timing in LCLKs(1)	11	6	2	1	0	0	8	1	0	15	6	2
	Enhanced IDE Spec in ns(2)	N/S	N/S	N/S	70	25	N/S	215	50	25	NS	NS	NS
Enhanced Mode: PCIIDE 43h [7:6], [5:4], [3:2], or [1:0]	Bit values in hex	0	0	0	1	2	2	0	1	2	0	0	0
DRDY: 1F6h/176h[3:1]	Bit values in hex	0	0	0	0	0	0	0	0	0	0	0	0
	Timing in LCLKs(1)	2	2	2	2	2	2	2	2	2	2	2	2
Cycle Time	Timing in LCLKs	20	13	8	6	5	4	17	5	4	32	16	8
	Enhanced IDE Spec in ns(2)	600	383	240	180	120	N/S	480	150	120	960	480	240

N/S = Not Specified, N/A = Not Applicable

(1) The actual timing (in LCLKs) that will be generated by the IDE controller if the recommended bit values in hex are programmed.

(2) The timing (in ns) as specified in the Enhanced IDE Specification.

N/S = Not Specified, N/A = Not Applicable

Table 15 16-Bit Timing Parameters with 25MHz PCI Bus

		IDE Transfer Modes											
		PIO Modes						Multi-Word DMA Modes			Single-Word DMA Modes		
Parameter: Register Bits	**Dimension**	**0**	**1**	**2**	**3**	**4**	**5**	**0**	**1**	**2**	**0**	**1**	**2**
Address Setup: 1F6h/176h[5:4]	Bit values in hex	1	1	0	0	0	0	0	0	0	0	0	0
	Timing in LCLKs(1)	2	2	1	1	1	1	1	1	1	1	1	1
	Enhanced IDE Spec in ns(2)	70	50	30	30	25	N/S	N/A	N/A	N/A	N/A	N/A	N/A
R/W Command Pulse: 1F0h/170h/1F1h/ 171h[7:4], Index-0/1	Bit values in hex	4	3	2	2	1	1	5	2	1	D	6	3
	Timing in LCLKs(1)	5	4	3	3	2	2	6	3	2	13	7	4
	Enhanced IDE Spec in ns(2)	165	125	100	80	70	N/S	215	80	70	480	240	120
R/W Recovery Time: 1F0h/170h/1F1h/ 171h[3:0], Index-0/1	Bit values in hex	6	2	0	0	0	0	4	0	0	8	2	0
	Timing in LCLKs(1)	8	4	2	1	0	0	6	1	0	10	4	1
	Enhanced IDE Spec in ns(2)	N/S	N/S	N/S	70	25	N/S	215	50	25	NS	NS	NS
Enhanced Mode: PCIIDE 43h [7:6], [5:4], [3:2], or [1:0]	Bit values in hex	0	0	0	1	2	2	0	1	2	0	0	1
DRDY: 1F6h/176h[3:1]	Bit values in hex	0	0	0	0	0	0	0	0	0	0	0	0
	Timing in LCLKs(1)	2	2	2	2	2	2	2	2	2	2	2	2
Cycle Time	Timing in LCLKs	15	10	6	5	4	3	13	4	3	24	12	6
	Enhanced IDE Spec in ns(2)	600	383	240	180	120	N/S	480	150	120	960	480	240

(1) The actual timing (in LCLKs) that will be generated by the MIDE Module if the recommended bit values in hex are programmed.

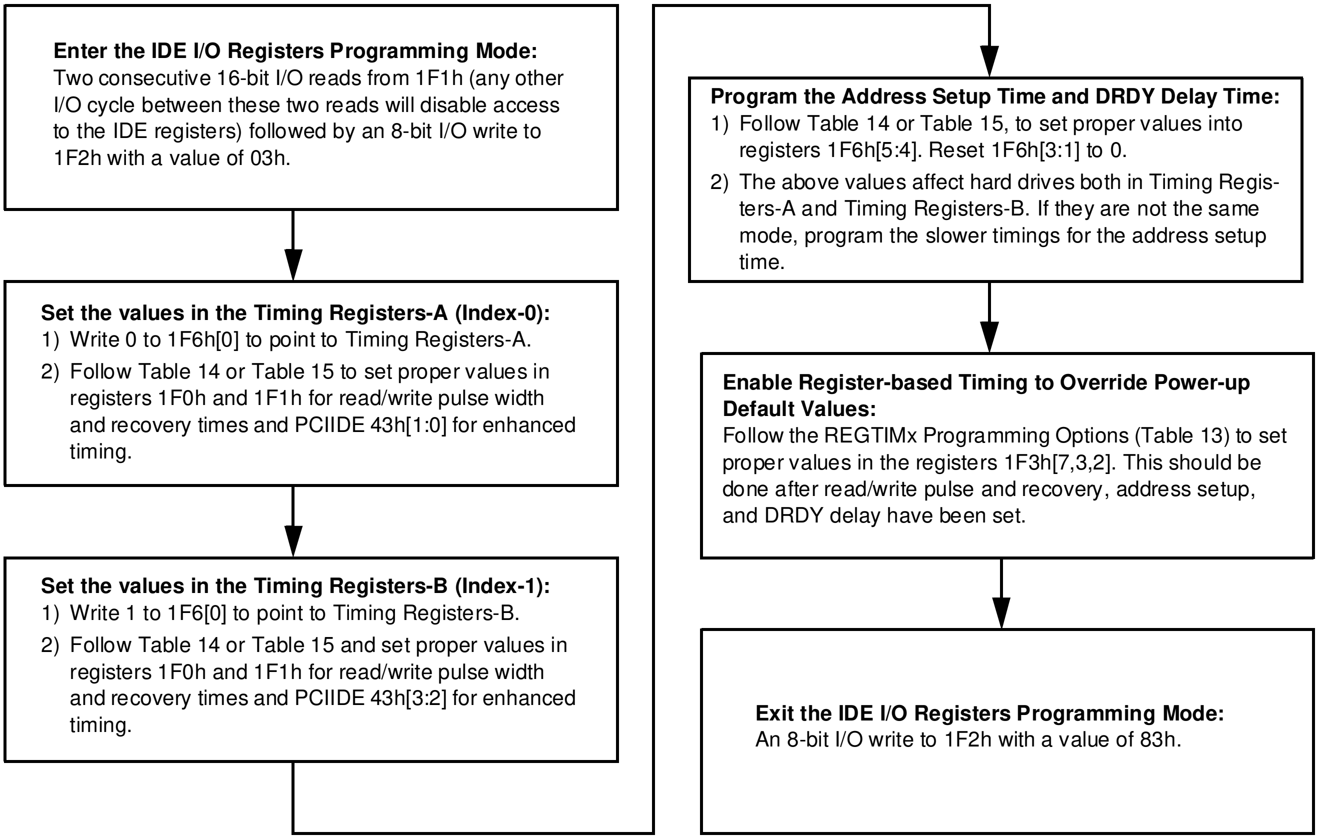
(2) The timing (in ns) as specified in the Enhanced IDE Specification.

Table 16 IDE Interrupt Routing Chart

		82C578 PCI Configuration Register Setting					Interrupt Controller			
Functions		**PCI Bus Intrf, Dev = 01h Func = 0**	**IDE Module, Dev = 01h Func = 1**				**82C578 Interrupt Input**		**IDE Interrupts Output**	
IDE Modes		**4Fh[6]**	**04h[0]**	**40h[3]**	**40h[2]**	**09h[3:0]**	**Pin 135[1]**	**Pin 136[1]**	**Primary**	**Secondary**
Primary	Secondary	IDE Module Enable	IDE I/O Enable	2nd IDE Disable	Native Mode Enable	Native/ Legacy Mode	IRQ14 or DINT0	IRQ15 or DINT1	8259 or PCI INCT	8259 or PCI INCT
Disable		0	PCI Config. Register Space cannot be accessed				ISA IRQ14	ISA IRQ15	N/A	N/A
		1	0	x	x	xxxx				
Legacy[2]	Disable	1	1	1	0	xxxx	DINT0	ISA IRQ15	8259 IRQ14	N/A
		1	1	1	1	xx10				
Native	Disable	1	1	1	1	xx11	DINT0	ISA IRQ15	PIRQ3#[3]	N/A
Legacy[2]	Native	1	1	0	1	1110	DINT0	DINT1	8259 IRQ14	PIRQ3#[3]
Native	Legacy[2]	1	1	0	1	1011	DINT0	DINT1	PIRQ3#[3]	8259 IRQ15
Legacy[2]	Legacy[2]	1	1	0	0	xxxx	DINT0	DINT1	8259 IRQ14	8259 IRQ15
		1	1	0	1	1010				
Native	Native	1	1	0	1	1111	DINT0	DINT1	PIRQ3#[3]	PIRQ3#[3]

1. The ISA IRQ14 (ISA IRQ15) will not be available to the ISA bus if the on-board primary (secondary) IDE is enabled.
2. The 8259 IRQ14 (8259 IRQ15) will not be available for PIRQ[3:0]# if the on-board primary (secondary) IDE is enabled.
3. In Native mode, IDE interrupts are shared with PIRQ3# from the PCI bus. It is routed in the same way as PIRQ3# to the interrupt controller and is controlled by PCIDV1 40h[11:9], 42h[7:1], and 50h[7:6] of the 82C578 (Device #01h, Function #0). Using this mode requires that the IDE device's Interrupt Service Routine support interrupt sharing.

Figure 2 IDE Interface Primary Channel Programming Flow Chart



Enter the IDE I/O Registers Programming Mode:
Two consecutive 16-bit I/O reads from 1F1h (any other I/O cycle between these two reads will disable access to the IDE registers) followed by an 8-bit I/O write to 1F2h with a value of 03h.

Set the values in the Timing Registers-A (Index-0):

1) Write 0 to 1F6h[0] to point to Timing Registers-A.
2) Follow Table 14 or Table 15 to set proper values in registers 1F0h and 1F1h for read/write pulse width and recovery times and PCIIDE 43h[1:0] for enhanced timing.

Set the values in the Timing Registers-B (Index-1):

1) Write 1 to 1F6[0] to point to Timing Registers-B.
2) Follow Table 14 or Table 15 and set proper values in registers 1F0h and 1F1h for read/write pulse width and recovery times and PCIIDE 43h[3:2] for enhanced timing.

Program the Address Setup Time and DRDY Delay Time:

1) Follow Table 14 or Table 15, to set proper values into registers 1F6h[5:4]. Reset 1F6h[3:1] to 0.
2) The above values affect hard drives both in Timing Registers-A and Timing Registers-B. If they are not the same mode, program the slower timings for the address setup time.

Enable Register-based Timing to Override Power-up Default Values:
Follow the REGTIMx Programming Options (Table 13) to set proper values in the registers 1F3h[7,3,2]. This should be done after read/write pulse and recovery, address setup, and DRDY delay have been set.

Exit the IDE I/O Registers Programming Mode:
An 8-bit I/O write to 1F2h with a value of 83h.

Programming and Drive Placement Tips:

1. Ensure that IDE I/O Address 1F6h[0] (176h[0] in the Secondary channel) is set to 0 whenever accessing Timing Registers-A. It is a common mistake that after accessing Timing Registers-B, this bit is not reset to 0 by the BIOS. An error happens after a soft reset (those bits will not be reset during a soft reset). The BIOS wants to reload the timing sets to both Timing Registers-A and -B. It would actually write to Timing Registers-B twice.
2. The address setup and recovery time are shared by the two IDE devices on the same channel at 1F6h[5:1]. If these two devices are not in the same mode, slower address setup and recovery time should be programmed to ensure proper timings on the slower drive. Under this assumption, two drives should be placed on the separate channels in a two-drive system. In a multiple-drive system, place slower drives on one channel and faster drives on the other channel.
3. If no IDE hard drives are in the primary slave, secondary master location or slave location, set only the command pulse and recovery time (1F0h/1F1h, Index-1, 170h/171h, Index-0 and 170h/171h, Index-1) to correspond to PIO Mode 0. This is to ensure proper timing for an ATAPI CD-ROM that may be in any of these locations.

System Configuration

This section of the document will discuss configuration of the I/Os in the Viper Xpress+ Chipset.

Cache Sizing Programming Guide

The following tasks need to be completed in order to size the cache subsystem.

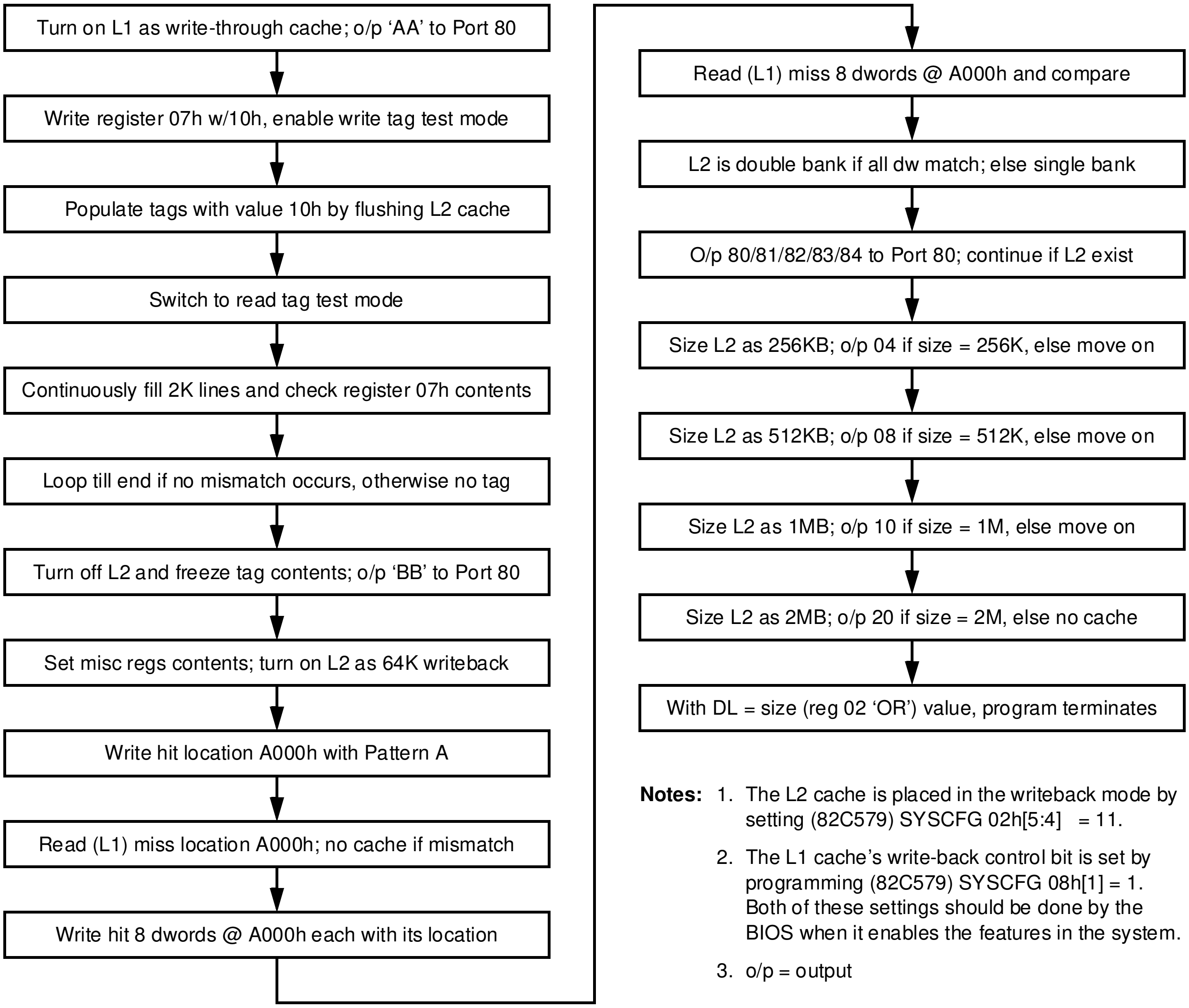
1. Performing cache tag test
2. Single bank cache SRAM devices detection

If a usable cache SRAM device is detected, then the cache needs to be sized. The following are valid cache sizes for the Viper Xpress+ Chipset.

- 256KB total cache size
- 512KB total cache size
- 1MB total cache size
- 2MB total cache size

The following is a flow diagram of the Viper Xpress+ Chipset cache sizing program.

Figure 3 Viper Xpress+ Chipset Cache Sizing Program


Turn on L1 as write-through cache; o/p 'AA' to Port 80
Write register 07h w/10h, enable write tag test mode
Populate tags with value 10h by flushing L2 cache
Switch to read tag test mode
Continuously fill 2K lines and check register 07h contents
Loop till end if no mismatch occurs, otherwise no tag
Turn off L2 and freeze tag contents; o/p 'BB' to Port 80
Set misc regs contents; turn on L2 as 64K writeback
Write hit location A000h with Pattern A
Read (L1) miss location A000h; no cache if mismatch
Write hit 8 dwords @ A000h each with its location
Read (L1) miss 8 dwords @ A000h and compare
L2 is double bank if all dw match; else single bank
O/p 80/81/82/83/84 to Port 80; continue if L2 exist
Size L2 as 256KB; o/p 04 if size = 256K, else move on
Size L2 as 512KB; o/p 08 if size = 512K, else move on
Size L2 as 1MB; o/p 10 if size = 1M, else move on
Size L2 as 2MB; o/p 20 if size = 2M, else no cache
With DL = size (reg 02 'OR') value, program terminates

Notes:

1. The L2 cache is placed in the writeback mode by setting (82C579) SYSCFG 02h[5:4] = 11.
2. The L1 cache's write-back control bit is set by programming (82C579) SYSCFG 08h[1] = 1. Both of these settings should be done by the BIOS when it enables the features in the system.
3. o/p = output

PCI Interrupt Routing

The registers that affect the assignment of interrupts in the Viper Xpress+ chipset are located in the 82C578. The registers that affect the assignment are PCIDV1 40h, 41h and 50h.

Table 17 Interrupt Assignment / Control

PCI Interrupt	PCIDV1	Bit Values/Triggered IRQ
PIRQ0#	40h[2:0]	000 = IRQx 001 = IRQ5 010 = IRQ9 011 = IRQ10 100 = IRQ11 101 = IRQ12 110 = IRQ14 111 = IRQ15
PIRQ1#	40h[5:3]	
PIRQ2#	40h[7:6], 41h[0]	
PIRQ3#	41h[3:1]	

Note: The bits provided above control the interrupt routing only when the bits are programmed to a value different from 000. In the event that the bits are programmed to 000 for a given PIRQx#, then the routing of that interrupt is controlled by PCIDV1 50h. That mapping scheme is provided in Table 18.

Table 18 Enhanced Interrupt Assignment / Control

PCI Interrupt	PCIDV1	Bit Values/Triggered IRQ
PIRQ0#	50h[1:0]	00 = Disable 01 = IRQ3 10 = IRQ4 11 = IRQ7
PIRQ1#	50h[3:2]	
PIRQ2#	50h[5:4]	
PIRQ3#	50h[7:6]	

Interrupt Sensing

Whenever PCI interrupts are routed to ISA interrupts, the interrupt needs to become level-triggered instead of edge-triggered. The only exception is the IDE interrupt (IRQ14). Table 19 lists the register bits that affect the interrupt trigger mechanism and the associated interrupt.

Table 19 Interrupt Level Assignment

ISA Interrupt	PCIDV1	Bit Value
IRQ3	51h[2]	0 = Edge-triggered 1 = Level-triggered
IRQ4	51h[1]	
IRQ5	42h[1]	
IRQ6	52h[7]	
IRQ7	51h[0]	
IRQ9	42h[2]	
IRQ10	42h[3]	
IRQ11	42h[4]	
IRQ12	42h[5]	
IRQ14	42h[6]	
IRQ15	42h[7]	

Enabling Features in the Viper Xpress+ Silicon

All features listed in this section of the document should be turned on by the BIOS while enabling the advanced features in the system. Where indicated, the BIOS needs to follow the order specified while turning on these features.

Note: Make sure writes to registers PCIDV0 44h-47h are always 32-bit writes. No memory access should be allowed in between configuration writes.

- CPU address pipelining
  - SYSCFG 08h[2] = 1
- PCI pre-snoop
  - The BIOS should turn on SYSCFG 0Fh[7] first and then turn on SYSCFG 16h[3]. This order should be maintained. As long as this order is maintained no other ordering with regard to the other features needs to be followed.
- Page miss posted write
  - SYSCFG 11h[2] = 1
- Hidden refresh (82C568)
  - PCIDV1 47h[6] = 1
- Programming wait states during PCI master transfers
  - Enhanced Mode - PCI master read:
    PCIDV0 44h[1] = 1
    PCIDV1 54h[6] = 1
    = Controlled by SYSCFG 20h[1:0]
    - SYSCFG 20h[1:0]
      01 = X-3-3-3
      10 = X-2-2-2
      11 = X-1-1-1
  - Enhanced Mode - PCI master write:
    PCIDV0 44h[2] = 1
    PCIDV1 54h[7] = 1
    = Controlled SYSCFG 20h[3:2]
    - SYSCFG 20h[3:2]
      01 = X-3-3-3
      10 = X-2-2-2
      11 = X-1-1-1

Note: It is highly recommended to program X-1-1-1 transfers for PCI masters

- CPU-to-DRAM FIFO
  - **Enable** (Sequence must be followed)
    - SYSCFG 01h[2] = 1 (CAS width for DRAM write - 2 CLK)
    - SYSCFG 02h[0] = 1 (CAS precharge - 1 CLK)
    - SYSCFG 02h[1] = 1 (DRAM posted writes if already enabled this bit will be set to 1)
    - SYSCFG 2Ch[1:0] = 11 (Generate BOFF when FIFO full)
    - PCIDV0 44h[4] = 1
    - PCIDV0 45h[1] = 1
  - **Disable** - Follow reverse sequence.
    Note: If DRAM post write already enabled there is no need to disable it when turning off this feature. Similarly for CAS width and precharge.
- PCI-to-DRAM FIFO
  - **Enable**/**Disable** (Sequence must be followed)
    - SYSCFG 20h[3:0] = 0Fh (PCI X-1-1-1 enable)
    - PCIDV0 47h[5] = 1
    - PCIDV0 44h[2:1] = 00
    - PCIDV0 44h[6:5] =
      00 - Disable read/write FIFO
      01 - Disable read/Enable write FIFO
      10 - Enable read/Disable write FIFO
      11 - Enable read/write FIFO
    - SYSCFG 2Ah[3:2] = 11
      (bit 2 enables reads burst)
      (bit 3 enables write burst)
- CPU-to-PCI FIFO
  - **Enable** (Sequence must be followed)
    - SYSCFG 15h[5:4] = 01 (if PCI posted writes are already enabled, this bit will be set to either 01, 10 or 11)
    - PCIDV0 44h[7] = 1
    - SYSCFG 2Eh[3] = 1
  - **Disable** - Follow reverse sequence.
    Note: If PCI post write already enabled do not disable it when turning off this feature.
- EDO 5-2-2-2 timing
  - Enable (Sequence must be followed)
    - Turn on EDO functionality
    - SYSCFG 1Dh[4] = 1 (Turns on 7-2-2-2)
    - SYSCFG 1Fh[4] = 1 (Turns on 6-2-2-2)
    - SYSCFG 26h[3] = 1
    - SYSCFG 2Dh[6] = 1
    - SYSCFG 2Dh[5:0] = Set for EDO bank detected (5-2-2-2)

- Self Refresh
  - **Enable**
    - SYSCFG 27h[2:0] =
100 - if ext clock is 66MHz
101 - if ext clock is 60MHz
110 - if ext clock is 50MHz
111 - if ext clock is 40MHz
    - PCIDV1 54h[0] = 1
  - **Disable**
    - PCIDV1 54h[0] = 0
    - SYSCFG 27h[2:0] = 000

Note: It is recommended to turn on self refresh and turn off hidden refresh.

- CPU to Sync SRAM 3-1-1-1-1 pipelining
  - **Enable** (Sequence must be followed)
    - SYSCFG 10h[5] = 1
    - SYSCFG 04h[3] = 1
- Fast NA (3 CLK single cycle writes)
  - Enable (Sequence must be followed)
    - Ensure SYSCFG 0Ch[6] =1
    - No L2 cache:
SYSCFG 0Eh[2] = 1
    - L2 cache enabled
SYSCFG 27h[4] = 1
SYSCFG 0Fh[4] = 1
PCIDV1 55h[1] = 1
PCIDV0 42h[0] = 1
SYSCFG 0Eh[2] = 1
- Buffer DMA
  - ISA Retry
SYSCFG 55h[1] = 1
SYSCFG 1Eh[3] = 1
SYSCFG 26h[6] = 1
SYSCFG 22h[4] = 1
- SDRAM
  - Read around
SYSCFG 2Ch[4] = 1
PCIDV0 44h[13:12] = 11
  - SDRAM timing
    - 9-1-1-1
SYSCFG 1Fh[4] = 0
SYSCFG 1Dh[4] = 1
PCIDV0 54h[5:0] = 111111
    - 8-1-1-1
SYSCFG 1Fh[4] = 0
SYSCFG 1Dh[4] = 1
PCIDV0 54h[5:0] = 000000
    - 7-1-1-1
SYSCFG 1Fh[4] = 1
SYSCFG 1Dh[4] = 1
PCIDV0 54h[5:0] = 000000
  - SDRAM pipeline
    - X-1-1-1/2-1-1-1
pipeline setup for EDO
SYSCFG 00h[6] = 1
SYSCFG29h[7] = 1
    - X-1-1-1/5-1-1-1
pipeline setup for EDO

Note: For all SDRAM-based systems, the following bits need to be set:
PCIDV0 48h[4] = 1,
PCIDV0 48h[3] = 0,
PCIDV0 4Eh[6] = 0
PCIDV0 4Eh[5] = 1
PCIDV0 55h[7] = 1
PCIDV0 55h[6] = 1

Deep Buffer Programming

The following sequence should be used when enabling deep buffers.

1. CPU-to-PCI
2. CPU-to-DRAM
3. PCI-to-DRAM

Also the enabling of deep buffers with respect to the entire BIOS, the following sequence is recommended.

1. The three deep buffers can be enabled just before giving control to boot block to start the OS.
2. When deep buffers are enabled and Ctrl+Alt+Del key is pressed (soft boot), the BIOS must disable all the three deep buffers in the reverse order of enabling after giving sufficient time (250 CPU clock time at least) to allow flushing of buffered data from the buffers. Again just before giving control to boot block, they should be enabled.
3. All the deep buffer enabling/disabling code must be a tight sequence code without any DRAM or non-configuration PCI cycles in-between.